Distributed Load Demand Scheduling in Smart Grid to Minimize Electricity Generation Cost

Siyu Yue, Di Zhu, Yanzhi Wang, and Massoud Pedram
Department of Electrical Engineering
University of Southern California, Los Angeles, CA, USA
{siyuyue, dizhu, yanzhiwa, pedram}@usc.edu

***Abstract*—Load demand scheduling of electricity consumers is an effective way to alleviate the peak power demand on the electricity grid and to combat the mismatch between generation and consumption. In this paper, we consider a scenario where multiple users cooperate to perform load demand scheduling in order to minimize the electricity generation cost. With the help of a central controller in the grid, a globally optimal solution can be achieved. However, this centralized solution may not always be feasible since it requires a huge amount of communication and the grid may not be equipped with such a central controller at all. Therefore, we propose a distributed load demand scheduling algorithm where each end user schedules its own tasks based on the partial information provided by other users. Simulation results show that this distributed load demand scheduling is able to achieve near-optimal solutions that has very little performance degradation compared to the centralized method.**

I. INTRODUCTION

One serious challenge we face in the electrical grid is the mismatch between electricity consumption and generation. On the consumption side, electric demand ramps up significantly during certain hours of a day (a.k.a. peak hours), and this peak consumption is increasing rapidly. On the generation side, it is very cost-ineffective for the utility companies to enhance their generation capacity to meet the increasing peak need. However, most of the generation capacity is wasted during off-peak hours. Typically in United States, the energy consumption is greater than 90% of the generation capacity in only 5% of the time [3].

An effective method to avoid the significant expense of increasing the generation capacity is to perform *demand side management* (DSM) [1], which aims at matching the consumers' electricity demand with the generation profile or reducing the peak demand. In order to incentivize end users to perform DSM, utility companies typically employ different pricing policies with higher unit price during peak hours and lower unit price during off-peak hours [4][5]. The DSM is mutually beneficial as the end users lower their electricity bills and the utilities companies reduce electricity generation cost.

There are two prevalent methods to perform DSM: load demand scheduling and energy storage system deployment. The former performs DSM by shifting end users' load task from peak hours (with higher unit price) to off-peak hours (with lower unit price). The latter equips home users with energy storage systems, which get charged during off-peak hours and supply power to the load during peak hours. In this way the power demand on the grid is also shifted away from the high-price hours effectively.

The most recent technology innovations in the field of smart grid provides new opportunities for DSM [2]. The construction of the communication infrastructure connects together all the generation facilities and household appliances in a certain area, and allows them to efficiently and promptly exchange information. Thanks to this communication network, both generation facilities and end users are more responsive to each other, and DSM can be performed in a more proactive manner.

In the current electricity grid, most end users only care about their own expenses. However, there are a couple of cases where the end users are cooperative. For example, when all the end users are in the same micro-grid and the electric energy generation facilities of this micro-grid are jointly owned by these users. The goal for the end users are thusly to minimize the cost to generate electricity to satisfy their demand. In addition, cooperation is also needed when a group of end users belongs to a single financial entity [9]. Another scenario is proposed by Mohsenian-Rad, *et al.* in [14], where all the end users, even when each of them pays for its own electricity bill, will become cooperative under a carefully designed pricing policy.

In this paper, we look at the DSM problem of a smart-grid system consisting of multiple cooperative end users and an electricity energy provider. The end users are cooperative since they share the same goal which is to minimize the overall electricity generation cost. A globally optimal solution can be achieved with a central controller deployed in the grid. However, the centralized approach may not be feasible as it has huge communication requirements: each user needs to send the information of their tasks (e.g., power demand profile, arrival time, deadline, etc.) to the central controller, and after scheduling, the controller sends back the detailed scheduling information of each task. Moreover, in some cases the grid may not have such a central controller. To address this issue, we propose an effective iteration-based distributed load scheduling algorithm that can be implemented in each end user. In each iteration every user schedules its own tasks and then broadcast its updated power demand profile to other users. The distributed algorithm converges very fast and has comparable performance with the centralized solution.

The rest of this paper is organized as follows. Section II discusses the related work on DSM. In Section III we present the system model. The DSM problem is then formulated mathematically in Section IV. The proposed distributed scheduling algorithm is proposed in Section V. Section VI demonstrates the simulation results and Section VII concludes the paper.

II. RELATED WORK

Load demand scheduling is one of the methods to reduce the electricity bills of end users. Mohsenian-Rad *et al.* in [6] and Conejo *et al.* in [7] formulate the energy consumption optimization problem of one single user and solve the problem based on effective predictions of future electricity prices. Hatami *et al.* in [9] consider a group of users sharing a single electricity bill and solved the problem of minimizing the bill under quasi-dynamic pricing policies for both continuous-time case and time-slot based case. Goudarzi *et al.* in [10] solve a similar problem by designing branch and bound-based exact algorithms and force-directed heuristic algorithms. Caron *et al.* in [8] consider a multi-user system under dynamic pricing policy and analyzed different scenarios based on the amount of information available to each user.

In addition to load task scheduling, DSM can also be done by incorporating electrical energy storage systems in residential households. Chiu *et al.* in [11] propose a management policy for the energy storage systems based on the current unit energy price and take system uncertainty and the physical constraints of the storage system into account. Wu *et al.* in [12] utilize the electric vehicles as distributed storages and employ a game theoretic approach to achieve optimal frequency regulation. In [13] Zhu *et al.* focus on the design of a residential storage system and propose a methodology to maximize the profit over the system's lifetime. Wang *et al.* in [14] propose a hierarchical control algorithm for managing storage systems based on the prediction of load consumption and PV power generation.

There are also some research papers that investigate how the utility companies can take advantage of DSM to maximize their profit or to minimize the electricity generation cost. In [15], Mohsenian-Rad *et al.* propose a game theoretic model for optimizing the electricity generation cost. The Nash equilibrium of this game is proved to be unique and optimal for all users. Yue *et al.* in [16] propose a dual-pricing policy for utility companies to flatten the aggregated power demand profile. Cui *et al.* study how utility companies can maximize their profits in a oligopolistic energy market in [17] and propose a game theoretic model to determine electricity prices for non-cooperative utility companies in [18].

III. SYSTEM MODEL

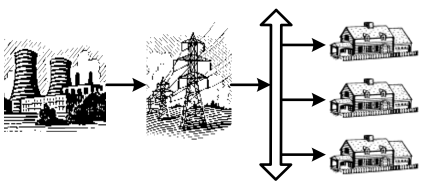

Figure 1. System architecture.

As shown in Figure 1, the system model in this paper consists of an electric energy provider (the power supply and the grid) and a group of cooperative end users. The electric energy provider supplies electricity to all the end users and they share the electricity bill which is the same as or proportional to the electricity generation cost. The end users are cooperative as they all have the same objective: to minimize the total electricity generation cost.

The detailed model of the end users' energy demand and the electricity generation cost is presented below.

A. Load Task Model

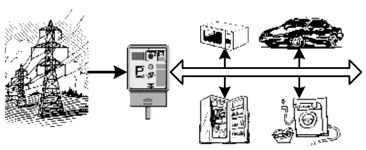

Figure 2. Home system structure.

In the proposed system, each user may be a single family house, an unit in a multi-family house, or an office building, etc. Each user has a number of electrical appliances connected to the energy provider via a smart meter, as shown in Figure 2.

We define a task as the usage of an electrical appliance for a certain period of time. Each user u_i has a set of tasks $\{T_{i,j}\}$ to be executed. The following parameters are used to describe each task $T_{i,j}$:

- $a_{i,j}$: the arrival time (i.e., the earliest time to start executing the task);
- $d_{i,j}$: the deadline (i.e., by which time the task must be finished);
- $l_{i,j}$: the duration (length);
- $p_{i,j}(t)$: the power demand profile, where $0 \le t < l_{i,j}$.

For a task $T_{i,j}$, all the parameters are given *a priori*. The DSM algorithm needs to determine the start time $s_{i,j}$ of each task to minimize the total energy cost. $s_{i,j}$ should conform to the arrival time and deadline constraint:

$$a_{i,j} \le s_{i,j} \le d_{i,j} - l_{i,j}$$

Obviously, a task can be scheduled for execution at different start times if and only if the duration of a task satisfies:

$$l_{i,j} < d_{i,j} - a_{i,j}$$

Note that in our system model we consider a time span of one day and divide it into multiple time slots.

B. Electricity Generation Cost

We denote the cost to generate the amount of energy E at time slot t as $C_t(E)$. This cost function may be different for different time slots. Assuming that the total energy consumption during time slot t is E_t, we can calculate the total electricity generation cost for one day by

$$\sum_t C_t(E_t)$$

The cost function is usually convex due to the increasing marginal cost for the energy provider to generate electricity.

IV. PROBLEM FORMULATION

Given the above system model, we formulate the multiple cooperative end users' load scheduling problem (referred to as MCLS hereinafter) as follows.

Given:

- The electricity generation cost function of each time slot $C_t(E)$;
- A group of cooperative users $\{u_i\}, i = 1,2, \dots, k$, where k is the total number of users;
- The task set of each user $\{T_{i,j}\}$, where $T_{i,j}$ indicates the j-th task of user u_i. $j \in \{1,2, \dots, N_i\}$ and N_i is the total number of tasks of user u_i;
- The parameters of each task: arrival time $a_{i,j}$, deadline $d_{i,j}$, duration $l_{i,j}$, and power demand profile $p_{i,j}(t)$.

Find:

- The start time $s_{i,j}$ of each task for every user.

Minimize:

- The overall electricity generation cost:

$$\sum_t C_t(E_t)$$

where

$$E_t = \sum_i \sum_j \hat{p}_{i,j}(t - s_{i,j})$$

and

$$\hat{p}_{i,j}(t - s_{i,j}) = \begin{cases} p_{i,j}(t - s_{i,j}), & \text{if } 0 \le t - s_{i,j} < l_{i,j} \\ 0, & \text{otherwise} \end{cases}$$

Subject to:

- The scheduling constraints of each task:

$$a_{i,j} \le s_{i,j} \le d_{i,j} - l_{i,j}$$

We prove that the MCLS problem is NP-complete (NPC).

Proof:

1) The solution to the MCLS problem can be verified in polynomial time (MCLS $\in NP$):

It is obvious that we can verify the solution using the above equations in polynomial time. Therefore, MCLS $\in NP$.

2) For a NPC problem L, L is polynomial reducible to the MCLS problem:

We adopt a well-known NPC problem, the set partitioning problem [19] in our proof. For a given set of N numbers $\{n_i\}$, a set partitioning problem L is to partition the set into two subsets S_1 and S_2 so as to minimize the difference between the sum of numbers in each subset, calculated by

$$\left|\sum_{i\in S_1} n_i - \sum_{i\in S_2} n_i\right|$$

To reduce the set partitioning problem to a MCLS problem, we consider two time slots and one user. Assume this user has N tasks. For all the tasks of this user, the arrival time is zero, the deadline is the end of the second time slot, the duration is one time slot, and the power demand of each task equals n_i. In this case, each task can be scheduled to either the first time slot or the second one. The amounts of energy consumption in the two slots are calculated by

$$E_1 = \sum_{s_i=1} n_i, E_2 = \sum_{s_i=2} n_i$$

Assume the cost functions are convex and the same for each time slot. Because of the convexity of the cost functions, the overall electricity generation cost is minimized when the difference between E_1 and E_2 is minimized. The optimal task scheduling of this MCLS problem is the same as the optimal set partition of the problem L.

Combining 1) & 2), the MCLS problem is NP-complete. ■

The NP-completeness of the proposed MCLS problem implies that it cannot be solved optimally with polynomial time complexity, thereby motivating us to find effective heuristic algorithms that can achieve near-optimal solutions with acceptable complexity.

The MCLS problem can be solved in either a centralized or a distributed fashion. In a centralized method, there is a central controller in the electricity grid. All users send their tasks' specification to the central controller at the beginning of a day. The central controller then finds out the scheduling solution (i.e., the start time of the users' tasks) and send the solution back to the end users. In a distributed method, each user determines the best scheduling solution for its own tasks, based on the information provided by other end users.

Centralized and distributed methods both have pros and cons. Centralized methods tend to achieve better solutions because the central controller has the complete information of all the end users' tasks. However, they require a large amount of communication since each user has to send to the controller the information of all its tasks. Moreover, when there is no such central controller in the electricity grid, a centralized method will be infeasible. On the contrary, distributed methods require less amount of communication and are able to harness the computing power distributed at each end user.

While this paper focuses on designing a good distributed scheduling algorithm, a centralized scheduling algorithm is proposed as the performance upper bound to evaluate the distributed scheduling algorithm.

V. Proposed Scheduling Algorithm

This section first presents a centralized scheduling algorithm for the system as the performance upper bound, and then proposes an efficient distributed algorithm.

A. *Centralized Scheduling Algorithm*

We adopt search-based simulated annealing algorithm for the centralized DSM. Simulated annealing algorithm is an effective heuristic algorithm for finding a good approximation of the global optimal solution. At each iteration of the simulated annealing, we define a *move* as follows: We find a neighboring state by randomly select one task with more than one feasible start time (i.e., $a_{i,j} < d_{i,j} - l_{i,j}$) and change its start time $s_{i,j}$ to a new feasible start time (i.e., $a_{i,j} \le s'_{i,j} \le d_{i,j} - l_{i,j}$ and $s_{i,j} \ne s'_{i,j}$). The cost c' of the new power demand profile is calculated and compared to the current cost c. If $c' < c$, meaning this move is a downhill move (which results in less total cost), we accept this move, and continue to execute the next iteration. Otherwise, i.e., in case of an uphill move, the probability of making this move q is given by the acceptance probability function, defined by

$$q = e^{-(c'-c)/K}$$

Since $c' \ge c$ in the case of an uphill move, the acceptance probability $0 < q < 1$. It decreases with the decrease of K, a.k.a. the temperature in the simulated annealing algorithm. We gradually decrease the temperature K (a.k.a. cooling down) to make the solution converge. The minimum cost is recorded during the simulated annealing process. The pseudo code of the proposed algorithm is shown below.

Simulated Annealing-Based Centralized Algorithm

Initialize the task scheduling by setting the start time $s_{i,j} = a_{i,j}$ for all $i = 1,2,\dots,k$ and $j = 1,2,\dots,N_i$
Record minimum cost $c_{\min}$ as the cost of the initialized task scheduling
For each annealing step
    Randomly pick a new start time of a random task $s'_{i,j}$
    Calculated the cost of the new profile c'
    If $c' < c$
        Accept this move
        Check if the new cost is smaller than the minimum cost $c_{\min}$; if so, update $c_{\min}$
    Else
        Accept this move by probability $q = e^{-(c'-c)/T}$
    Endif
    Decrease temperature K
EndFor
Return $c_{\min}$, the minimum cost found during the process
EndAlgorithm

B. *Distributed Scheduling Algorithm*

We propose an iterative algorithm for the distributed scheduling method. Each iteration contains a scheduling phase and a communication phase. In the scheduling phase, each user schedules all its tasks to minimize the overall cost estimated by this user based on other users' power demand profiles. In the communication phase, each user broadcasts its updated power demand profile to other users according to the current scheduling of its own tasks. The iteration stops when the aggregated profile converges, i.e., when difference of the profiles achieved in two consecutive iterations is smaller than a threshold value.

A naïve approach for a user in the scheduling phase is to compute the best scheduling for each task of this user so that the overall energy cost is minimized assuming other users' power demand profile remains the same. However, this assumption is not valid since other users are simultaneously updating their scheduling during the same iteration as well. What is worse, this naïve approach may not converge as shown in the following example.

Assume there are two time slots and two users each with one task, as shown in Figure 3. The unit energy cost is a convex function of energy consumption in the time slot. One can easily tell that the optimal solution is to avoid collision of the two tasks, i.e., place one task in Time Slot 0 and another in Slot 1. But the naïve approach may result in an infinite loop.

Initially, the two tasks are both scheduled to Time Slot 0. In the first iteration, the users exchange information of their current scheduling results. Then in the scheduling phase, both users re-schedule their tasks to Time Slot 1 to avoid collision. However, it results in a new collision as both tasks are now in Time Slot 1. In the next iteration both tasks are rescheduled to Time Slot 0 and becomes an infinite loop.

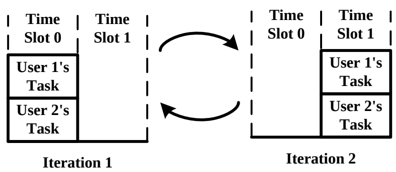

Figure 3. An example of infinite loop in the naïve approach.

We employ two techniques to overcome the above defect. The first technique is to use a exponential predictor to predict other users' power demand profile in the next iteration. Denote the prediction result of other users' power profile as $\boldsymbol{P_r}$ and the actual profile after scheduling in each iteration as $\boldsymbol{P_a}$. In each iteration we update the prediction by

$$\boldsymbol{P_r} \leftarrow \alpha \boldsymbol{P_r} + (1-\alpha)\boldsymbol{P_a}$$

where α is a parameter in the exponential predictor.

The second approach is to reschedule each task only with some probability instead of always rescheduling it to the place with minimum cost. The probability ε to reschedule task $T_{i,j}$ is:

$$\varepsilon = 1 - r\sum_t p_{i,j}(t)$$

where r is the reluctance coefficient. A task is less likely to be rescheduled with a greater value of r. In this way the probability of task collision is reduced. The algorithm starts with a small value of r to explore the possible optimal scheduling, and the convergence of the distributed algorithm can be guaranteed by gradually increasing r after each iteration.

The pseudo code for the distributed algorithm executed separately by each user is shown below:

Iterative Distributed Algorithm

Initialize other people's actual power demand profile to $\boldsymbol{P_a}$
Initialize power demand profile prediction $\boldsymbol{P_r}$ to zero
While difference of $\boldsymbol{P_a}$ between two iterations larger than th
    // Scheduling Phase
    Use $\boldsymbol{P_r}$ as other users' power demand profile
    For each task $T_{i,j}$
        For each $s_{i,j}$ from $a_{i,j}$ to $d_{i,j} - l_{i,j}$
            Calculate the electricity generation cost of the profile if task $T_{i,j}$ is scheduled to start at time $s_{i,j}$
        EndFor
        Find $s_{i,j}$ which minimizes the cost
        Update $s_{i,j}$ with a probability of ε
    EndFor
    // Communication Phase
    Broadcast the power demand profile to other users
    Upon receiving other users' profile, calculate $\boldsymbol{P_a}$
    Update prediction $\boldsymbol{P_r}$
    Increase the reluctance coefficient r
EndWhile
EndAlgorithm

VI. Simulation Results

A. Simulation Setup

In the following simulations, we set the number of time slots to be 48, which means that the length of each time slot is half an hour. We generate eight test cases using a stochastic model. The parameters of each task is derived from commonly used electrical appliances [20]. TABLE I. summarizes the number of users and the average number of tasks each user has for all eight test cases.

TABLE I. Specification of Each Test Case

Test Case #	Number of Users	Average Number of Tasks Per User
1	50	10
2	50	105
3	100	51
4	100	104
5	100	9
6	200	9
7	200	48
8	500	48

The electricity generation cost function we use for each time slot is a quadratic function:

$$C_t(E) = A_t E^2 + B_t E$$

The baseline algorithms are:

1) As-soon-as-possible algorithm (ASAP): each task is scheduled to start at its arrival time.

2) As-late-as-possible algorithm (ALAP): each task is scheduled so that it finishes right before deadline.

3) Centralized simulated annealing (SA) proposed in Section V.A.

4) Distributed scheduling without cooperation (NO-COOP): each user runs the simulated annealing algorithm to schedule its own tasks with no information on other users' tasks.

B. Performance Comparison

Figure 4 shows the power demand profile achieved by different algorithms for test case 3. As can be seen, the SA scheduling algorithm achieves a very desirable profile as its power demand during peak hours is flat. The profile of the proposed distributed algorithm is also much better than that of ASAP, ALAP and NO-COOP since it has a lower peak and less fluctuations. Figure 5 shows the performance of different load demand scheduling algorithms normalized to the SA algorithm's outcome. The centralized SA algorithm achieves the best performance in all test cases as expected. The performance of the proposed distributed algorithm outperforms the ASAP, ALAP and NO-COOP scheduling algorithm and is only 0.67% worse than that of the SA scheduling algorithm.

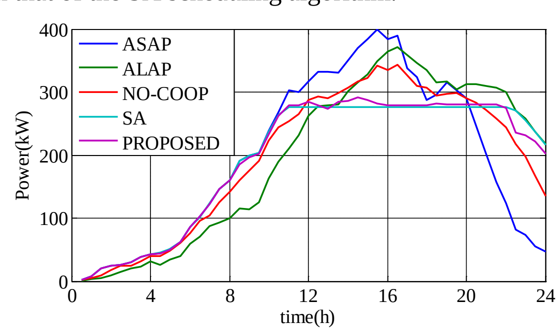

Figure 4. Power profiles of different algorithms for test case 3.

C. Convergence Speed

In the proposed distributed algorithm we use the reluctance coefficient to ensure convergence. TABLE II. shows the number of iterations the proposed algorithm takes for each test case. As can be seen in the table, the number of iterations are all very small and, more importantly, does not scale up with the number of users.

TABLE II. NUMBER OF ITERATIONS IN THE PROPOSED ALGORITHM

Test Case #	1	2	3	4	5	6	7	8
Number of Iterations	5	8	10	17	6	7	8	7

Figure 5 shows how the increment amount of the reluctance coefficient r influences the convergence speed and the performance of the proposed distributed algorithm (normalized to the SA algorithm's outcome). As shown in the figure, the number of iterations increases as the increment step size of the reluctance coefficient becomes smaller, but the performance of the distributed algorithm gets improved. When the increment step is small enough (0.01), the performance of the proposed algorithm is almost identical to the centralized SA algorithm.

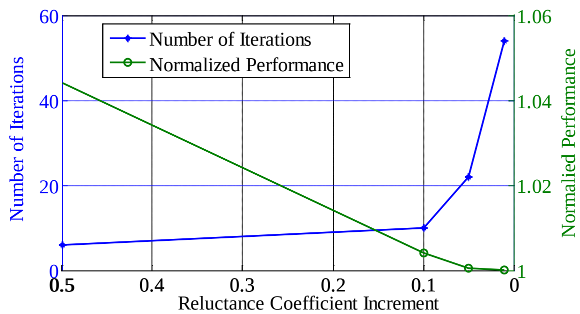

Figure 5. Convergence speed and performance as a function of the reluctance coefficient increment step size.

VII. CONCLUSION

In this paper, we consider the load demand scheduling problem of multiple cooperative end users. The objective of these end users is to minimize the electricity generation cost. Although a centralize method is able to achieve an optimal solution, it may not always be feasible due to high communication requirement and hardware overhead. Therefore we propose an iterative distributed load demand scheduling algorithm. The algorithm is conducted by each end user separately and the communication between different users only occurs at the end of each iteration. Simulation results show that the proposed distributed load demand scheduling algorithm converges fast and is able to achieve near-optimal results with very little performance degradation compared to the centralized method.

REFERENCES

[1] R.M. Delgado, "Demand-side management alternatives," *Proc. of the IEEE*, vol. 73, no. 10, pp. 1471-1488, October 1985

[2] F. Saffre, and R. Gedge, "Demand-side management for the smart grid," *2010 IEEE/IFIP Network Operations and Management Symposium*, pp. 300-303, April 2010

[3] E.M. Lightner, and S.E Widergren, "An orderly transition to a transformed electricity system," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 3-10, June 2010

[4] S. Borenstein, M. Jaske, and A. Rosenfeld, "Dynamic pricing, advanced metering and demand response in electricity markets," Center for the Study of Energy Markets, University of California at Berkeley, 2002

[5] H. Allcott, "Real time pricing and electricity markets," Harvard University, January 2009

[6] A.-H. Mohsenian-Rad, and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 120-133, September 2010

[7] A.J. Conejo, J.M. Morales, and L. Baringo, "Real-time demand response model," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 236-242, December 2010

[8] S. Caron, and G. Kesidis, "Incentive-based energy consumption scheduling algorithms for the smart grid," *2010 First IEEE International Conference on Smart Grid Communications*, pp. 391-396, October 2010

[9] S. Hatami, and M. Pedram, "Minimizing the electricity bill of cooperative users under a quasi-dynamic pricing model," *2010 IEEE International Conference on Smart Grid Communications*, October 2010

[10] H. Goudarzi, S. Hatami and M. Pedram, "Demand-side load scheduling incentivized by dynamic energy prices", 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011

[11] W. Chiu, H. Sun, and H.V. Poor, "Demand-side energy storage system management in smart grid," *2012 IEEE Third International Conference on Smart Grid Communications,*. Nov. 2012

[12] C. Wu, H. Mohsenian-Rad, and J. Huang, "Vehicle-to-Aggregator Interaction Game," *IEEE Transactions on Smart Grid*, Vol. 3, No.1, March 2012

[13] D. Zhu, Y. Wang, S. Yue, Q. Xie, M. Pedram, and N. Chang, "Maximizing return on investment of a grid-connected hybrid electrical energy storage system," *Proc. of Asia and South Pacific Design Automation Conference*, Jan. 2013

[14] Y. Wang, S. Yue, L. Kerofsky, S. Deshpande, and M. Pedram, "A hierarchical control algorithm for managing electrical energy storage systems in homes equipped with PV power generation," *Proc. of the IEEE Green Technologies Conference*, Apr. 2012

[15] A.-H. Mohsenian-Rad, V.W.S. Wong, J. Jatskevich, R. Schober and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 320-331, December 2010

[16] S. Yue, J. Chen, C. Ye, Y. Shi and Y, Gu, "Dual-pricing policy for controller-side strategies in demand side management," *2011 IEEE International Conference on Smart Grid Communications*, 2011

[17] T. Cui, Y. Wang, H. Goudarzi, S. Nazarian, and M. Pedram, "Profit Maximization for Utility Companies in an Oligopolistic Energy Market with Dynamic Prices," *IEEE Online Conference on Green Communications*, 2012

[18] T. Cui, Y. Wang, S. Yue, S. Nazarian, and M. Pedram, "A Game-Theoretic Price Determination Algorithm for Utility Companies Serving a Community in Smart Grid," *Proc. of IEEE PES Innovative Smart Grid Technologies Conference*, Feb. 2013

[19] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C, "Introduction to algorithms," *MIT press*, 2001

[20] http://www.energystar.gov/index.cfm?c=products.pr_find_es_products

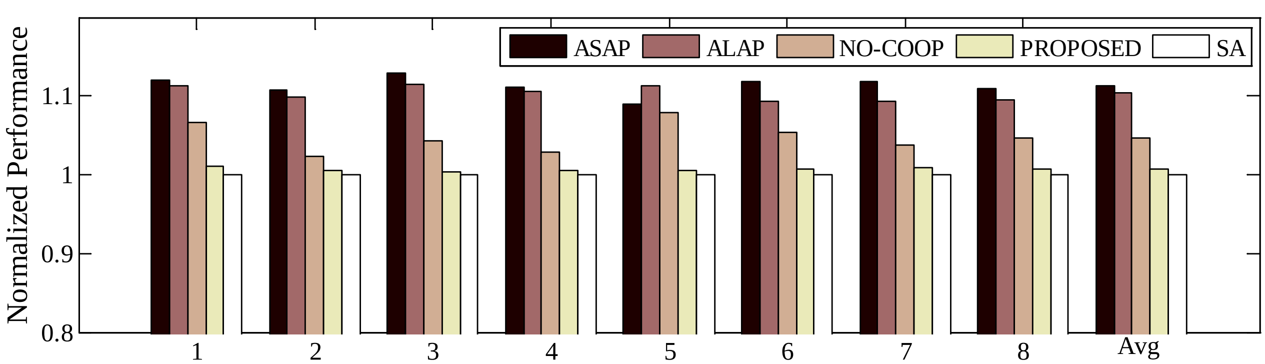

Figure 6. Normalized performance of different load demand scheduling algorithms.